Science Advances
AAAS

advances.sciencemag.org/cgi/content/full/6/26/eaax9070/DC1

Supplementary Materials for

The life history of human foraging: Cross-cultural and individual variation

Jeremy Koster*, Richard McElreath, Kim Hill, Douglas Yu, Glenn Shepard Jr., Nathalie van Vliet, Michael Gurven, Benjamin Trumble, Rebecca Bliege Bird, Douglas Bird, Brian Codding, Lauren Coad, Luis Pacheco-Cobos, Bruce Winterhalder, Karen Lupo, Dave Schmitt, Paul Sillitoe, Margaret Franzen, Michael Alvard, Vivek Venkataraman, Thomas Kraft, Kirk Endicott, Stephen Beckerman, Stuart A. Marks, Thomas Headland, Margaretha Pangau-Adam, Anders Siren, Karen Kramer, Russell Greaves, Victoria Reyes-García, Maximilien Guèze, Romain Duda, Álvaro Fernández-Llamazares, Sandrine Gallois, Lucentezza Napitupulu, Roy Ellen, John Ziker, Martin R. Nielsen, Elspeth Ready, Christopher Healey, Cody Ross

*Corresponding author. Email: jeremy.koster@uc.edu

Published 24 June 2020, *Sci. Adv.* **6**, eaax9070 (2020)
DOI: 10.1126/sciadv.aax9070

This PDF file includes:

SUPPLEMENTAL MATERIALS

1. Description of the data

The total sample contains 1,821 individual hunter, 23,747 hunter-level outcomes, and 21,160 trips across 40 study sites. To compile the dataset, the first author searched for relevant studies on subsistence hunting in the anthropological and biological literature, subsequently contacting authors to invite them to contribute data. The contributors submitted data in a standardized format that included variables for the biomass acquired on terrestrial hunting trips, the ages of the hunters at the time of the hunt, the duration of the trip, the hunting weaponry carried by the hunters, and the presence of dogs or assistants (the distinction between hunters and assistants was left to the discretion of contributors, who were counseled to conceptualize "hunters" as those individuals who made active contributions to detecting and pursuing prey). Our data are restricted to hunting, and exclude gathering, because of the paucity of data on gathered plant foods.

There is tremendous imbalance in sample size across units. One site contributes only 6 trips from 2 individuals. Another contributes more than 14,000 trips from 147 individuals. Some individuals contribute only a single outcome, while others contribute dozens. The majority of the sample comprises male hunters, with too little data on female hunters to infer generalizable sex differences. (This does not imply that men's production and skill is more relevant to human evolution, nor that women's foraging skill would necessarily exhibit either the same or a different functional relationship with age.) Most sites contribute primarily cross-sectional data, while a few others exhibit impressive time series. The statistical framework is designed to make use of all these data.

2. The life history foraging model

Since skill cannot be directly observed, what is required is a model with latent age-varying skill. This unobservable skill feeds into a production function for observable hunting returns. In this section, we define a framework that satisfies this requirement. We explain it one piece at a time, with a focus on the scientific justification.

One advantage of the latent skill approach is that it allows us to use different observations from different contexts—both solo and group hunting, for example—to infer a common underlying dimension of skill. But modeling even the simplest foraging data benefits from this approach, as hunting returns often are highly zero-augmented. Separate production functions for zeros and non-zeros are needed to describe such data. In principle, more than one dimension of latent skill could be modeled. We restrict ourselves to only one in the current analysis. With more detailed data, describing additional dimensions should be possible.

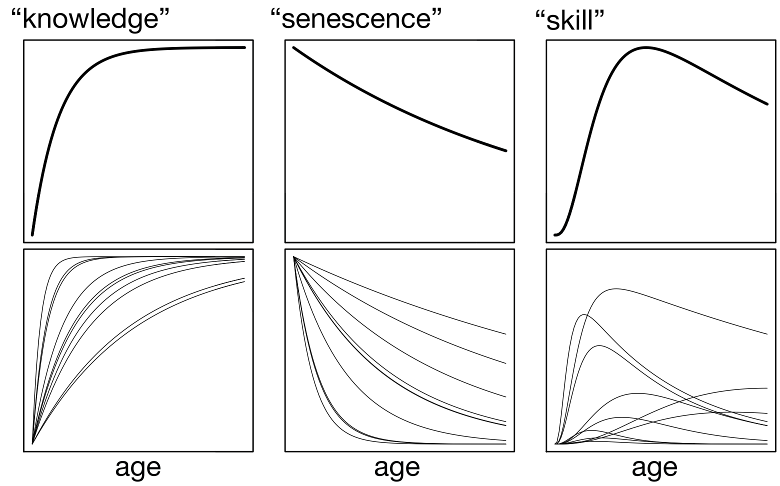

FIGURE S1. The age-specific skill model. Top row: Increasing components, “knowledge,” and decreasing components, “senescence,” multiply to produce relative productive potential at each age, “skill.” Bottom row: Variation in the components combines to produce a diverse array of possible skill functions.

2.1. **Latent skill model.** One of the simplest life history models is the von Bertalanffy asymptotic growth model. We use this model to represent the increasing components of hunting skill as a function of age. These increasing components include knowledge, strength, cognitive function, and many other aspects that contribute to hunting success and increase but decelerate with age. For convenience, label the composite of these components *knowledge*. Assume that the rate of change in knowledge with respect to age x is given by $\mathrm{d}K/\mathrm{d}x = k(1 - K(x))$. This means only that knowledge increases at a rate proportional to the remaining distance to the maximum—the more there is left to learn, the more one learns. Solving this differential equation yields the age-specific knowledge of a hunter at age x:

$$K(x) = 1 - \exp(-kx) \tag{1}$$

where $k > 0$ is a parameter that determines the rate of increase. To account for senescence, we assume that production capacity M declines at a constant rate, given by $\mathrm{d}M/\mathrm{d}x = -mM(x)$. Solving this yields:

$$M(x) = \exp(-mx) \tag{2}$$

where $m > 0$ represents the rate of decline. The total age-specific skill is given by a weighted product of these two functions:

$$S(x) = M(x)K(x)^b \tag{3}$$

where the parameter b controls the relative importance of K. In economic terms, b is the knowledge elasticity of skill. We assume that k and m may vary across individuals—some people learn faster or senesce more slowly—while b is a property of the production context at a given study site.

This model is among the simplest we can construct. Nevertheless, it is capable of describing diverse age-specific skill curves. Figure S1 illustrates the general shapes of each component of the model, as well as how variation in each component may produce variable life histories. Each plot in this figure shows age on the horizontal axes. The top row of the figure illustrates the general shape of each component (left and middle) and one possible resulting lifetime skill curve (right). The bottom row shows 10 different, randomly simulated knowledge and senescence curves, with their implied random skill curves. These demonstrate that even a model as simple as this one, with only three parameters, is nevertheless capable of producing many diverse age-specific curves. This approach brings two more advantages, as compared to the use of polynomial functions of age. First, the parameters have straightforward biological interpretations. Second, these functions do not exhibit instabilities such as Runge's phenomenon that complicate fitting and prediction.

These functions also have clear weaknesses. Neither the rate of gain k nor the rate of loss m is plausibly constant over large age ranges. The rate of variation in body growth, for example, will produce rate variability in skill growth. And near the end of life, skill loss should accelerate rather than slow down. Although the data analyzed in this paper do not span the age ranges in which this variation would occur, we should be cautious about overgeneralizing from this analysis.

The final component of the core skill model is partial pooling of information. Since these data contain repeat measurements on the same units—individuals and sites—as well as substantial imbalance in sampling of these units, partial pooling via multilevel modeling provides superior estimates. We employ two levels of hierarchical pooling (Figure S2). First, the life history parameters k and m are pooled across individuals within each site (left column, Figure S2). In standard terminology, k_{ID} and m_{ID} for each individual are random effects drawn from a bivariate distribution. Each site also has its own value for b, reflecting variation in the relative importance of knowledge across sites. Therefore each site has its own distribution of skill functions (middle column). Finally, the site distributions are pooled together to regularize inference at the second level (right column), producing a distribution of site distributions. To an extent, this global distribution is a statistical fiction that is necessary to pool information properly among sites. However, it is also a target of inference, providing a weighted summary of all of the evidence across sites. The weights arise from the structure of the multilevel model and are functions of the sample sizes, the differences in site means, and the variation among those means. For example, a site could have a large sample size but contribute little to the global mean if its own mean were extreme.

2.2. **Production model.** Skill is not directly observable. Rather, we must infer it by its effects on hunting productivity. This requires introducing a layer of production functions through which skill acts. The production data available to us contain two correlated components: (1) the probability of a successful trip that produces a non-zero harvest and (2) the size of harvests obtained on successful trips. We model each with a standard log-linear function of labor, skill, and technology. Specifically, for successful trips, the mean expected harvest at skill S is given by:

$$h(S) = S^{\eta_h} L^{\beta_h} \exp \alpha_h \tag{4}$$

where η_h is the elasticity of skill, which determines the magnitude of skill differences on harvest, L^{β_h} is the labor allocated with its elasticity β_h, and α_h is a linear model including terms for technology and cooperation variables. (In this equation, the h subscript denotes

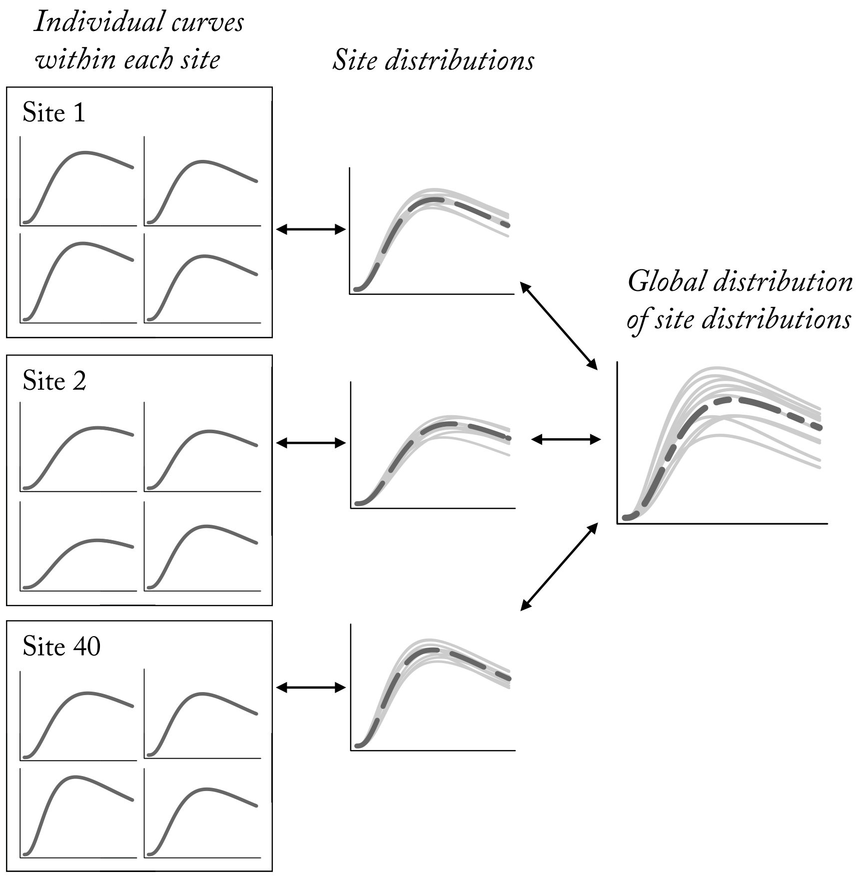

FIGURE S2. Hierarchical structure of skill functions within the inferential model. Within each site (left) a skill curve is inferred for each hunter. Individuals within each site are pooled using a distribution of individual skill curves (middle). Finally, the distributions of parameters within each site are again pooled using a distribution of distributions (right). This formulation allows variation among individuals to vary by site.

that this is the function for the non-zero harvests, as distinguished from the subsequent Bernoulli function for successes and failures in Equation 5, which features the p subscript.) Notice that harvest increases with both skill and labor, but that the elasticity of each determines the impact of any increase. The full distribution of harvests is assumed to follow a gamma distribution, which allows for the highly skewed distributions typical of many hunting data sets. However, a log-normal distribution of harvests would work as well. The important features are to impose a zero lower bound and to allow for positive skew. If we had detailed data on the encounters and pursuits of individual prey types, we could build a mixture distribution to better describe observed harvest sizes. But such data are available in very few cases. For comparability across sites and compatibility with the logit function described next (equation 5), we have proportionally standardized harvests relative to average harvest sizes at the respective study sites. When evaluating sources of variation in the data, it is important to bear in mind this standardization, which limits the inferences that can be made about between-site variation in this analysis.

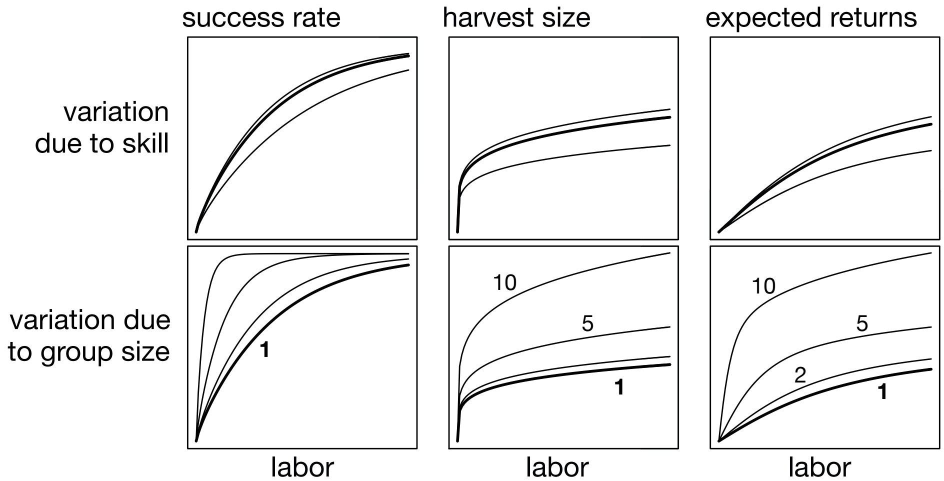

FIGURE S3. Example production functions for observed harvests. Expected harvest (righthand column) is the product of the probability of a non-zero harvest (lefthand column) and the expected size of a non-zero harvest (middle column). The top row shows how each component may vary with skill. The bottom row shows how each may vary with number of hunters.

A similar approach provides a Bernoulli distribution of success/failure. The probability that a trip produces a non-zero harvest is:

$$p(S) = 2\left(\text{logit}^{-1}(S^{\eta_p} L^{\beta_p} \exp \alpha_p) - \tfrac{1}{2}\right) \tag{5}$$

The terms enclosed within the interior parentheses recapitulate the log-linear production function of the above equation (4). The remainder of the function re-scales the log-linear model so that $p(S)$ varies continuously from zero to one and $p(0) = 0$.

This is a descriptive approach. It has the advantage of being able to describe many possible relationships between skill, labor, and technology. Figure S3 illustrates some of the model's features. Each plot in this figure shows labor input—hours allocated to foraging—on the horizontal axis. From left to right, the plots show the probability of a non-zero harvest, the expected harvest size on a successful trip, and the expected returns resulting from the product of the two. Each row illustrates the impact of one type of variation—variation in individual skill in the top row and variation in hunting group size in the bottom row. The first thing to notice is that the function implies monotonic returns to labor. Marginal returns must always either increase or decrease with labor. Second, skill and labor can influence hunting success and harvest size quite differently. There is no assumption that skill or labor is equally important for both components of production. And since technology can influence elasticity of skill and labor, technology can have independent effects as well.

2.3. **Cooperative trips and aggregated harvests.** Many of the hunting trips in our sample are cooperative, in the sense that multiple hunters of varying skill interact in producing returns. The harvests on these trips may be assignable to individual hunters or alternatively credited to the group as a whole. We handle cooperative trips by treating them as analogous to technology, with group size represented as a coefficient in the production equation.

When returns are aggregated to the level of the group rather than assigned to individual hunter, we replace individual hunter skill in the production equation with a weighted average of the skill of the group members.

2.4. **Missing values and measurement error.** Our sample embodies common statistical challenges. First, there are many missing values, notably for trip duration and the presence of dogs on trips. Second, there is measurement error, notably for individual ages. The customary solution to these problems is to drop all cases with any missing values and to replace uncertain measurements with their means. Instead of dropping cases with missing values, however, we model the unknown values. This allows Bayesian imputation of missing values, averaging over uncertainty in unobserved durations. We rely upon the same principle to handle measurement error in age. In some cases, co-authors who contributed datasets to our sample assigned a standard error to the recorded ages of hunters. Within the model, each hunter's date-of-birth is replaced with an unknown parameter with a prior centered on the recorded age and with standard deviation equal to the recorded standard error. In a few cases, no age is recorded for an individual. In those cases, we assign a vague prior that covers the entire range of observed ages.

2.5. **Inference.** The full model contains just under 28,000 parameters. Many of these correspond to missing durations and age uncertainties, and so contribute little fit to the sample. Many of the remaining parameters arise from the hierarchical structure of the life history model. These parameters do not make it easier to fit the sample, but rather harder. They reduce overfitting, by pooling information among sampling units. For the remaining parameters, we adopt regularizing priors that are more conservative than the implied flat priors of typical non-Bayesian procedures. We present a complete description of the priors in the supplemental code. Having fit alternative parameterizations of the model, we believe the results that we present in the next sections are qualitatively robust to changes in priors and even the hierarchical structure of the model. To facilitate alternative estimates of model parameters, though, we provide our annotated modeling code in this supplemental material (Appendix A), and we provide the full code and auxiliary scripts in the online supplements (https://osf.io/2kzb6/).

3. Formal model definition

As a complement to the above qualitative description of the modeling approach, we turn to a formal description of the model. Let y be an indicator variable for hunting success (produced a non-zero harvest) and h any observed non-zero harvest. Let i index observed outcomes (harvests). Then:

$$y_i \sim \text{Bernoulli}(p_i)$$
$$h_i \sim \text{Gamma}(\mu_{i2}, \nu_{\text{site}[i]})$$

The expressions for p and μ specify the production functions, indexed by j for the outcome type (for successes or harvest size, respectively):

$$p_i = 2\big(\text{logit}^{-1}(\mu_{i1}) - \tfrac{1}{2}\big)$$
$$\log(\mu_{ij}) = \eta_{\text{site}[i]j} S_{\text{trip}[i]} + \beta_{\text{site}[i]1j} \log L_i + \alpha_{ij}$$

The labor input is L_i, the duration of the trip, standardized so that the average trip at each site has $L = 1$.

The skill input S into the above is given by the average skill among the individuals contributing labor to a particular observed harvest:

$$S_{\text{trip}[i]} = n_{\text{trip}[i]}^{-1} \sum_{f=1}^{n_{\text{trip}}[i]} \exp(-m_{\text{id}[f]}\ell_{\text{id}[f],\text{trip}[i]})\big(1 - \exp(-k_{\text{id}[f]}\ell_{\text{id}[f],\text{trip}[i]})\big)^{b_{\text{site}[\text{trip}[i]]}}$$

where n is the number of productive foragers for trip$[i]$ (excluding individuals categorized as assistants, such as porters) and id$[f]$ is the forager ID of the f-th forager on each trip. This means that for aggregated harvests, in which individual contributions cannot be identified, the model uses average skill. The age $\ell_{f,\text{trip}[i]}$ is the estimated age for forager f at the time of trip$[i]$. We describe the age model further down. Note that all ages within the model are standardized by dividing calendar age by the reference age of 80, making $\ell = 1$ equivalent to 80 years old.

The intercept component of each production function, α_{ij}, is composed from:

- A site-specific intercept $a_{\text{site}[i]j}$
- A site-specific and outcome-type specific set of coefficients (elasticities) for the impact of group size, number of assistants, firearms, and dogs. The latter two variables are binary variables indicating whether the hunter had use of a gun (as opposed to other weaponry) or at least one dog.

On the log scale, these combine additively:

$$\alpha_{ij} = a_{\text{site}[i]j} + \mathit{groupsize} + \mathit{assistants} + \mathit{firearms} + \mathit{dogs}$$

All of these effects are allowed to vary by site as random effects. These assumptions are visible in precise detail in the statistical code.

Random effects on skill. The life history parameters k, m, and b make use of partial pooling both within and between sites. We use a two-level pooling structure that allows each site to have its own covariance between k and m. Specifically, let ID be the unique ID number of each forager. Then each k_{ID} and m_{ID} are defined by:

$$k_{\text{ID}} = \exp(W_1 + V_{\text{site}[\text{ID}[i]],1} + v_{\text{ID},1})$$
$$m_{\text{ID}} = \exp(W_2 + V_{\text{site}[\text{ID}[i]],2} + v_{\text{ID},2})$$

The parameters W_1 and W_2 are overall means, across all sites, and the parameters $V_{s,1}$ and $V_{s,2}$ are the offsets of these means for site s. This leaves $v_{\text{ID},1}$ and $v_{\text{ID},2}$ as the offsets for individual ID.

Starting at the lowest level, each pair of parameters $\mathrm{v}_{\text{ID}} = \{v_{\text{ID},1}, v_{\text{ID},2}\}$ are allocated probability from a bivariate normal:

$$\mathrm{v}_{\text{ID}} \sim \text{MVNormal}((0,0), \Sigma_{\text{site}[\text{ID}]})$$
$$\Sigma_S = \begin{pmatrix} \sigma_{S,1}^2 & \sigma_{S,1}\sigma_{S,2}\rho_S \\ \sigma_{S,1}\sigma_{S,2}\rho_S & \sigma_{S,2}^2 \end{pmatrix}$$

Each site is characterized by 6 parameters: site-level offsets for k, m, and b, as well as standard deviations for hunter-level k and m and their correlation ρ. These 6 parameters are themselves pooled across sites. This produces the distinction between variance among sites and the variance of the individual hunters, as described in the text.

Age error model. We accommodate uncertainty in observed ages by defining:

$$\ell_{\text{ID},i} = (\text{age}_i - \upsilon_{\text{ID}})/80$$
$$\upsilon_{\text{ID}} \sim \text{Normal}(l_{\text{ID}}, e_{\text{ID}})$$

where l_{ID} is the observed year of birth and e_{ID} is the assigned standard error. In the limit where $e_{\text{ID}} \to 0$, the age is purportedly known with certainty. Some sites reported ages using uniform intervals. We converted those to Gaussian representations with equivalent variances, so that the imputed ages were unconstrained. In most cases, when a researcher records a uniform age interval, they imply that the true age is closer to the middle of the interval and do not imply that it is impossible for the true age to be outside the interval. To allow this information into the model, we had to use something other than a uniform probability distribution. Gaussian is the most conservative choice, in that case. The irony of the effort put into dealing with age uncertainty is that it has no detectable impact on inference. Fixing all of the ages at their central value produces the same inferences that we reported in the main text. On the one hand, this is disappointing, because it really was not trivial to model the uncertainty, and it did not seem to matter much. On the other hand, it is important to do the right thing, even if it turns out not to matter.

4. Supplemental Results

4.1. **Production functions.** The skill functions presented in Figure 2 of the main text are inputs into site-specific production functions. These functions have their own elasticities and therefore characteristic shapes. Here we present alternatives to Figure 2 that illustrate these production functions. There are three different perspectives on the production function. The first component is the probability of success at each age. The second component is the distribution of harvest sizes at each age. These two components multiply to produce the distribution of expected harvests at each age.

To make these components easier to understand, consider all four implied components of the production function for only the Aché sample (Figure S4). The orange functions in the upper-left are the same latent skill functions as in the main text. The red functions in the upper-right are the probabilities of success for each hunter, with the horizontal dashed line showing 50% success rate. The points are the raw data—the proportion of successes at each observed age, aggregated across individuals who were observed at those ages. The lower-left blue functions are the expected harvest sizes, conditional on a non-zero harvest. Again the points are raw data—the average harvest observed at each age. The violet functions in the lower-right are just the product of the red and blue functions, showing the expected production at each age.

Each component may be of interest in itself. In some sites, such as the Aché (16 ACH), the success of each hunt contributes more to variation than does the harvest size. The red curves in Figure S4 vary more both across age and across individuals than do the blue curves. As a result, more of the variation in the resulting expected production curves, seen in violet, arises from success rates rather than variation in harvest sizes. As seen in the subsequent plots (Figure S5, Figure S6, Figure S7) the Matsigenka sample (9 MTS) shows the same pattern—more variation in success rates than harvest sizes. This is possibly a result of the prey types available at the respective sites. Regardless of the explanation, decomposing the expected production in this way shows how skill can influence some aspects more than others.

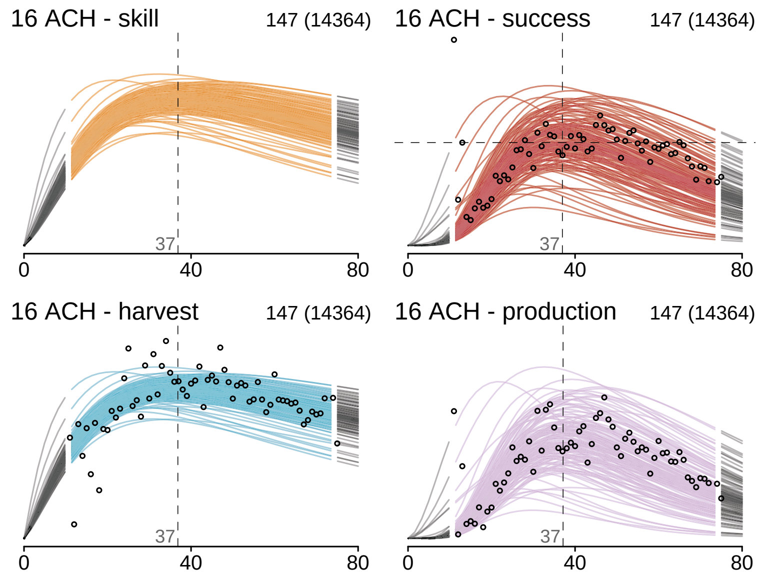

Figure S4. Components of the forager production functions for the Aché sample. See text for description.

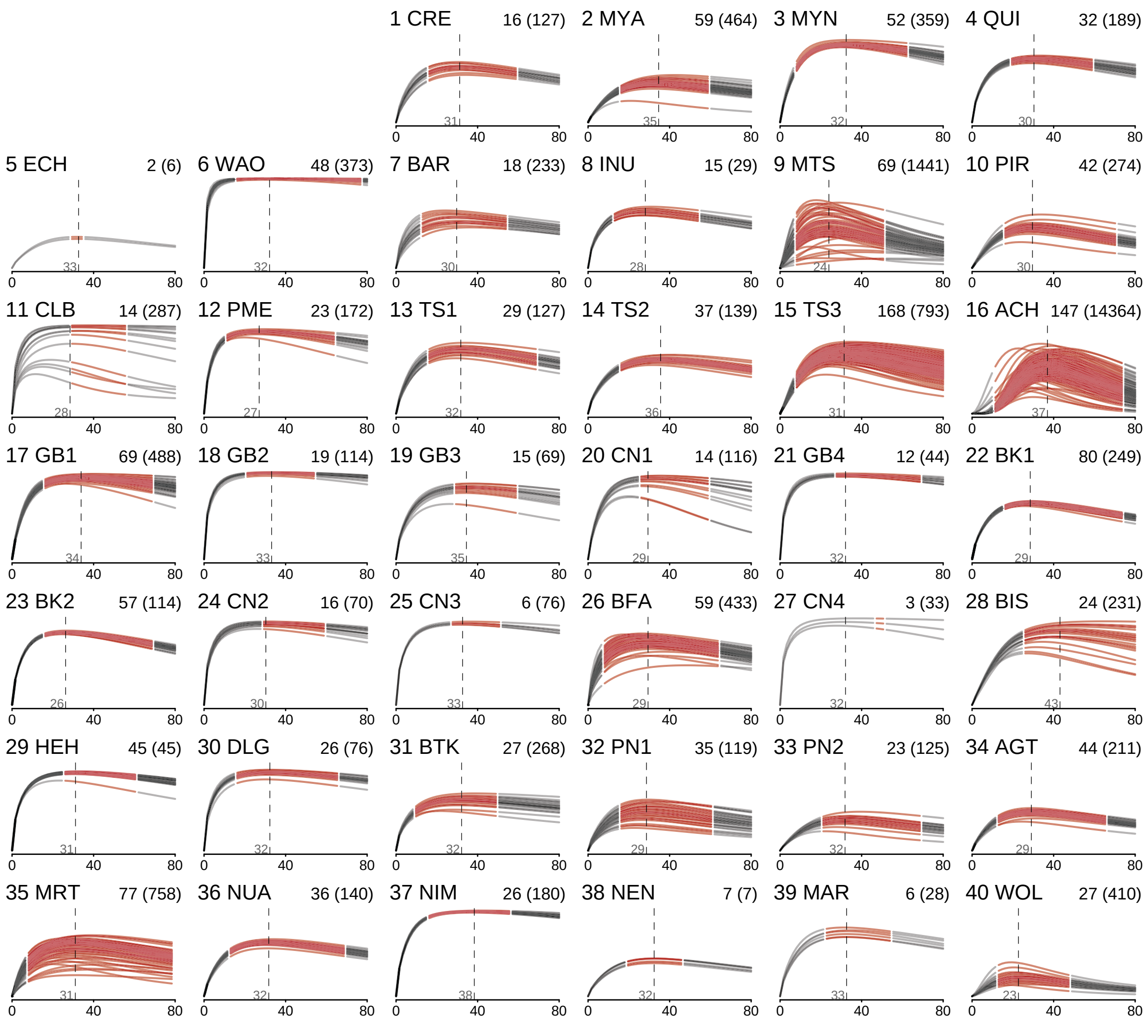

Figure S5. Posterior mean probabilities of hunting success across age. The axis ranges from 0 to 1, and is the probability of hunting success (a non-zero harvest). Model predictions are for a solitary excursion by a lone hunter without companions or assistants, and the predictions assume average hunt duration and the absence of dogs and firearms. Several sites, such as GB4 (21) and DLG (30), show essentially no variation in hunting success, since virtually all documented trips result in a non-zero harvest. Other sites, such as MRT (35) and WOL (40), show substantial failure rates and variation arising from it. nb: Variation in methods for documenting unsuccessful hunts imposes limitations on comparisons across sites – see the help files in the *cchunts* package for details.

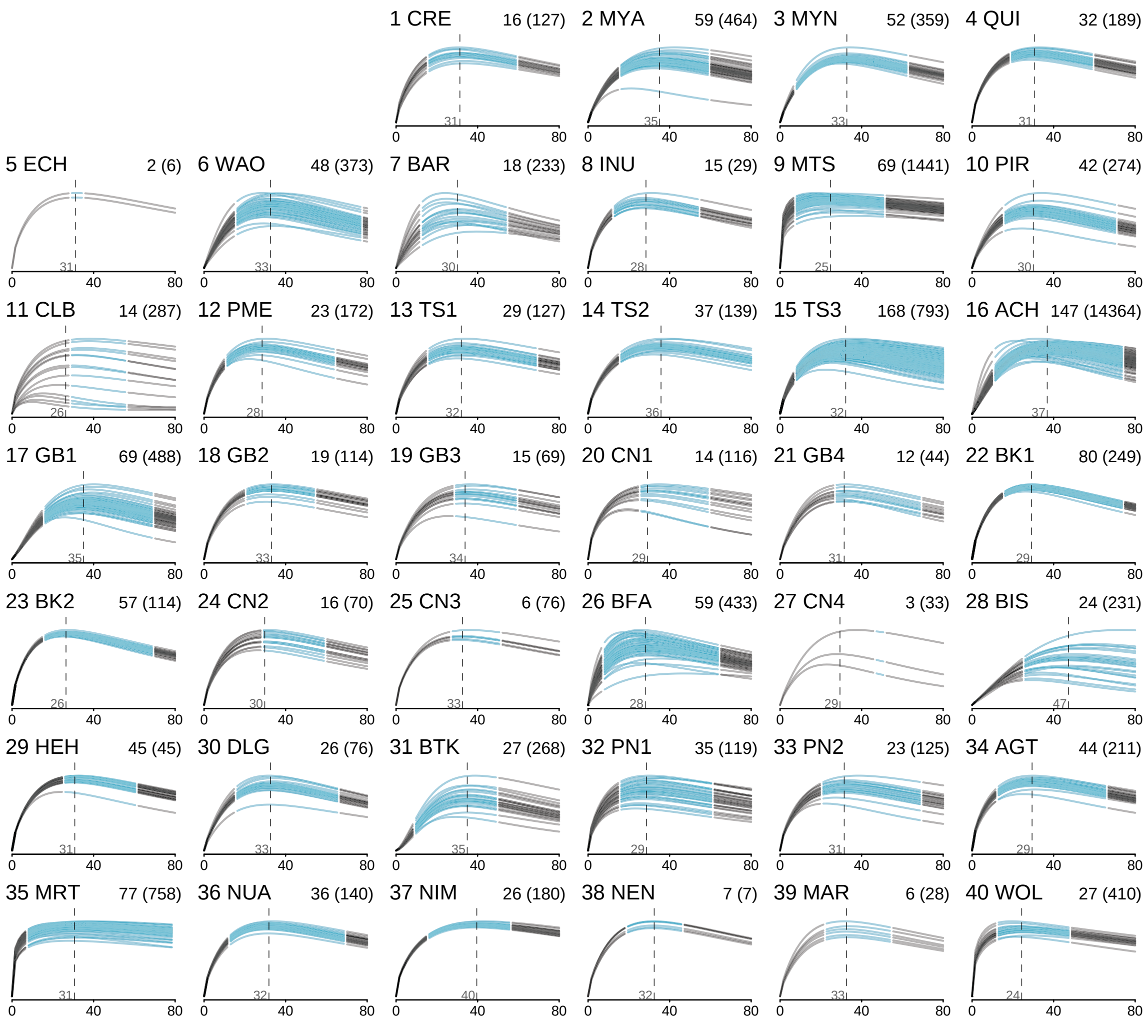

Figure S6. Posterior mean non-zero harvest size across age. Model predictions are for a solitary excursion by a lone hunter without companions or assistants, and the predictions assume average hunt duration and the absence of dogs and firearms. The vertical axis is proportion of maximum harvest at each site. While the units are not comparable across sites, therefore, hunter-level variation within sites is informative.

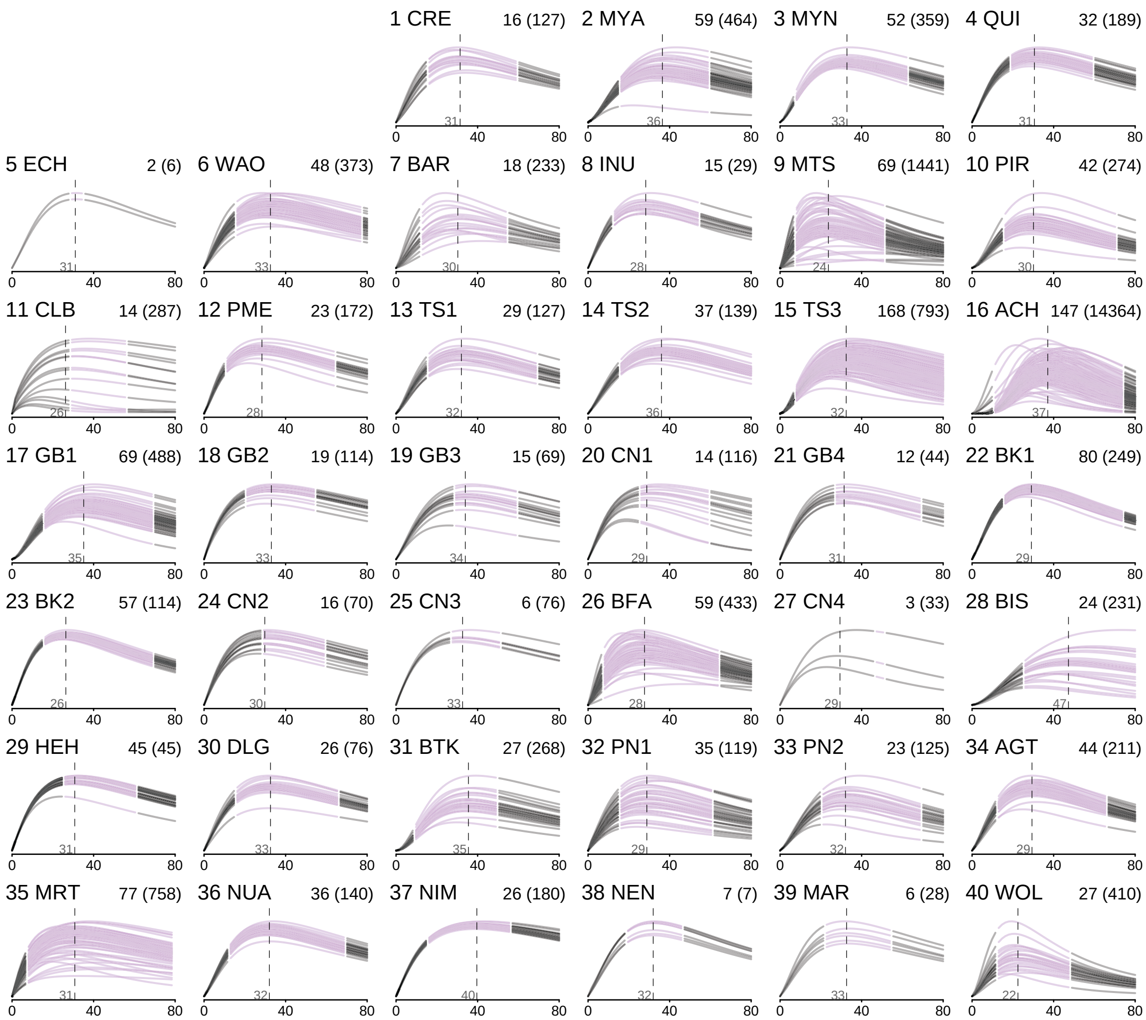

FIGURE S7. Expected production across age. These functions are the product of the success function and the expected harvest function. In considering relative expected energy contributions of individuals at different ages, these curves are perhaps the most relevant representations of the data.

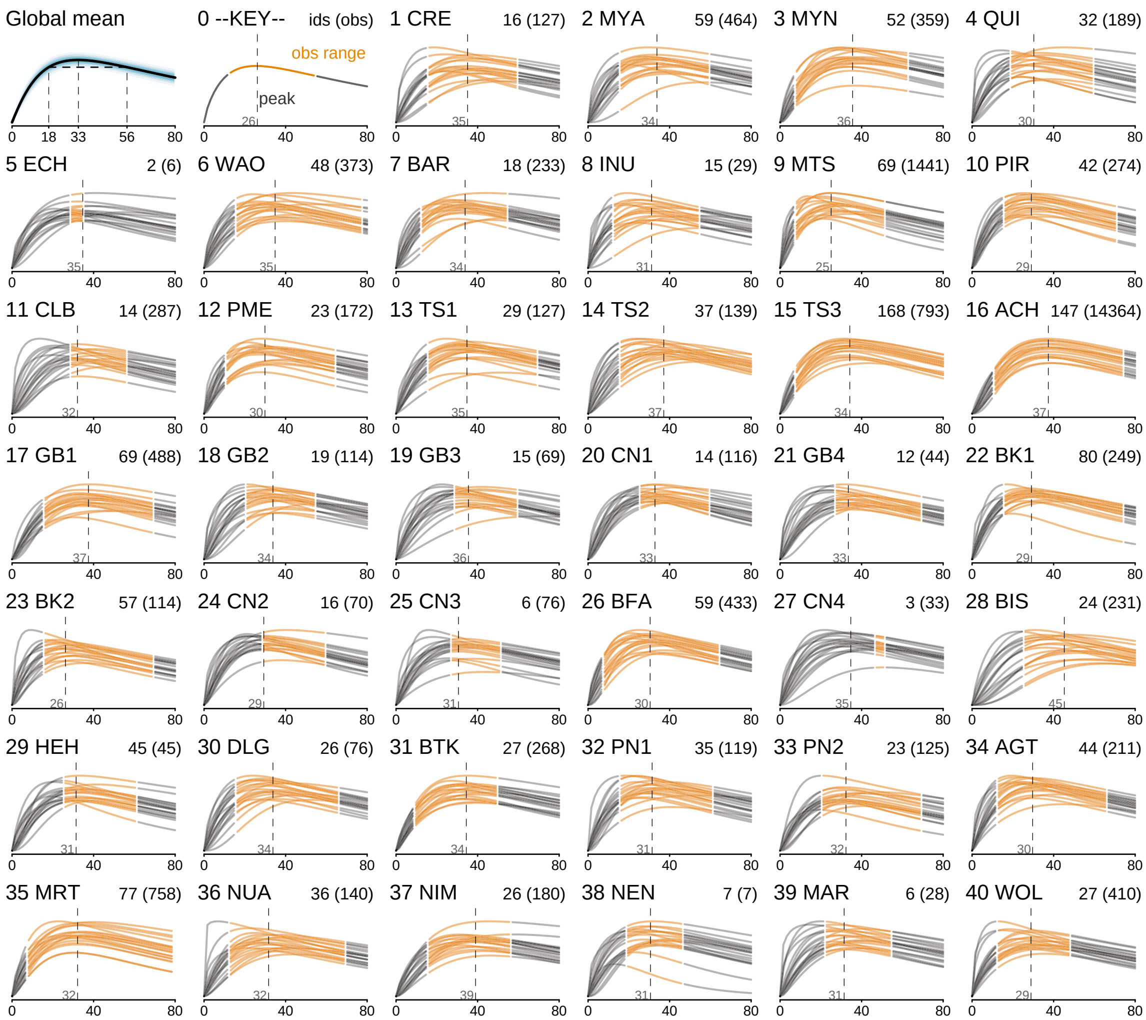

Figure S8. Simulated samples from the posterior distributions of skill functions in each site. This figure is similar to the skill grid in the main text, but it shows simulated hunters, not the posterior means for the observed hunters. This representation of the skill functions shows that, at many study sites, the model expects more empirical variation than can be seen in the previous skill figure (Figure 2).

Marginal posterior distributions. Many of the parameters in the production functions are interesting in themselves. For example, the marginal effects of group size and technology potentially inform debates about human subsistence strategies. In the figures that follow, we present marginal posterior distributions for all of these parameters, labeled informatively. In general, many of the parameters exhibit cross-cultural variation. For instance, in a small number of sites, the use of firearms or dogs increases the respective probabilities of a successful hunt or the amounts of biomass acquired. In most sites, however, the posterior distributions of these parameters are largely indistinguishable from the priors. (In some cases, this potentially reflects the lack of variation in the use of firearms or dogs within sites.) The elasticities of labor and skill inputs exhibit analogous cross-site variation.

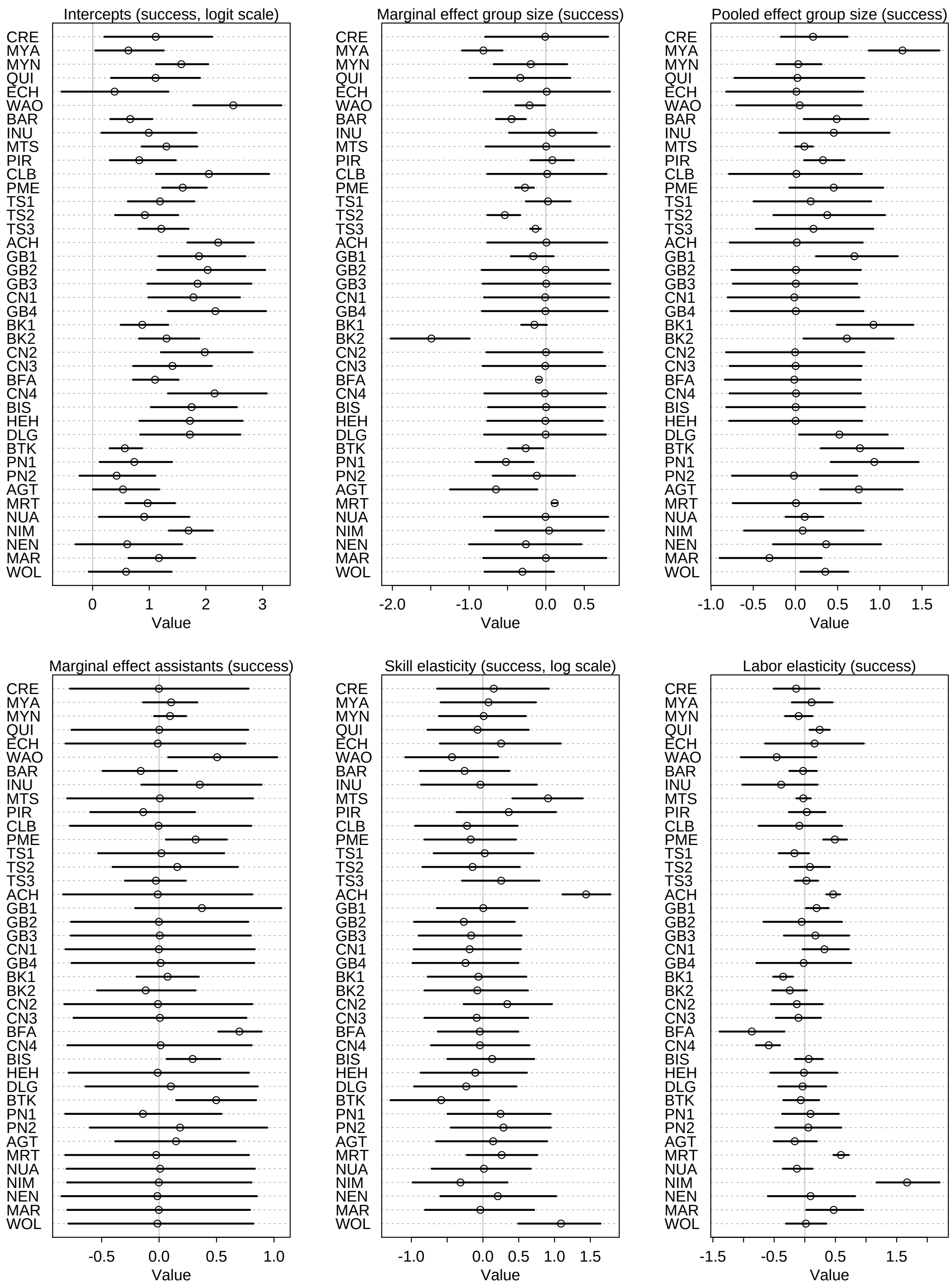

FIGURE S9. Marginal posterior distributions for production components (success). In the code, these parameters are named `af[1]`, `af[2]`, `af[3]`, `af[4]`, `sef`, and `bhours[1]`, respectively. Note that marginal distributions centered on zero with standard deviation 0.5 correspond to the prior. In those cases, the society contained no information to inform the parameter.

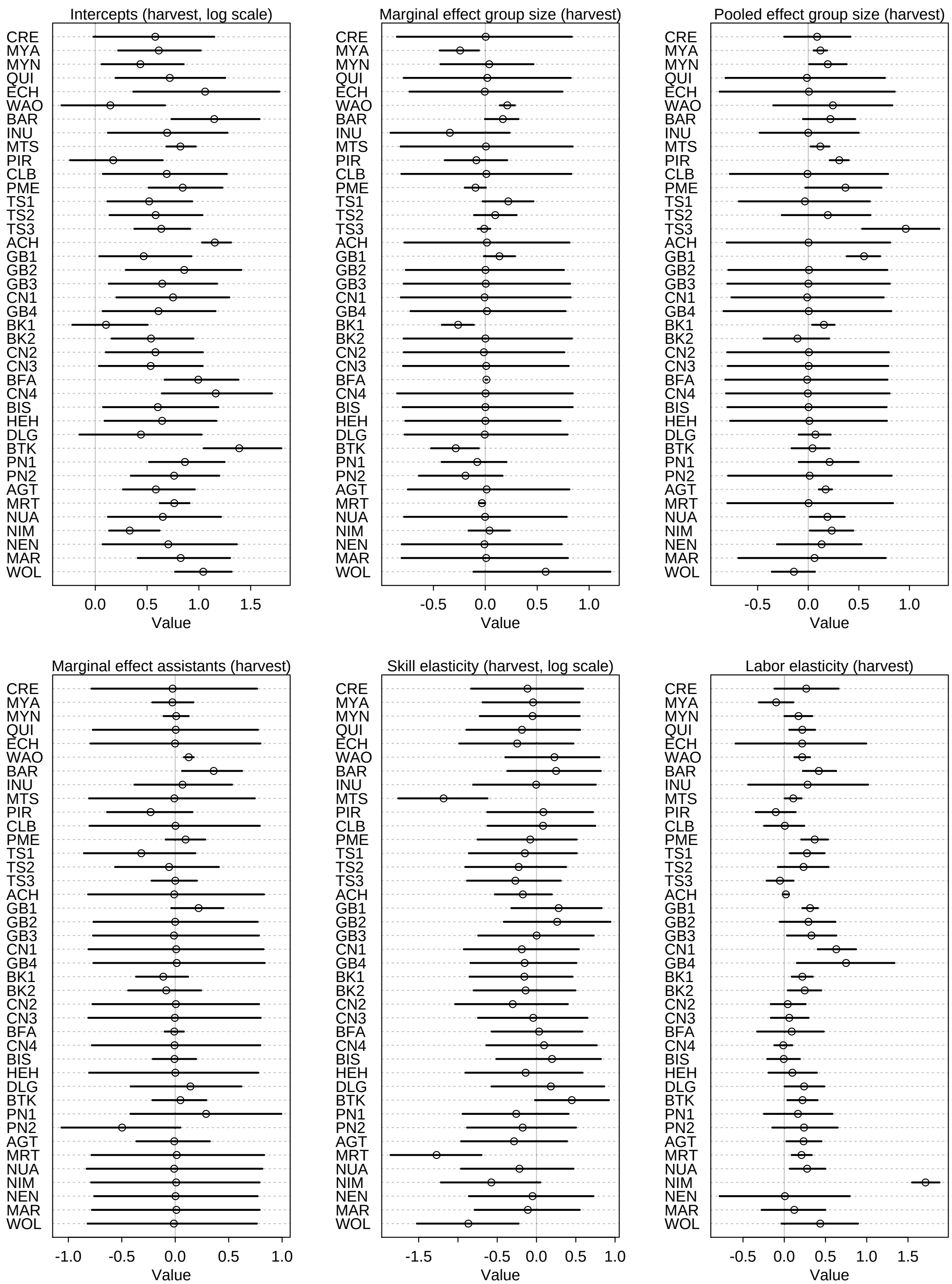

FIGURE S10. Marginal posterior distributions for production components (harvest). In the code, these parameters are named `ah[1]`, `ah[2]`, `ah[3]`, `ah[4]`, `seh`, and `bhours[2]`, respectively. Note that marginal distributions centered on zero with standard deviation 0.5 correspond to the prior. In those cases, the society contained no information to inform the parameter.

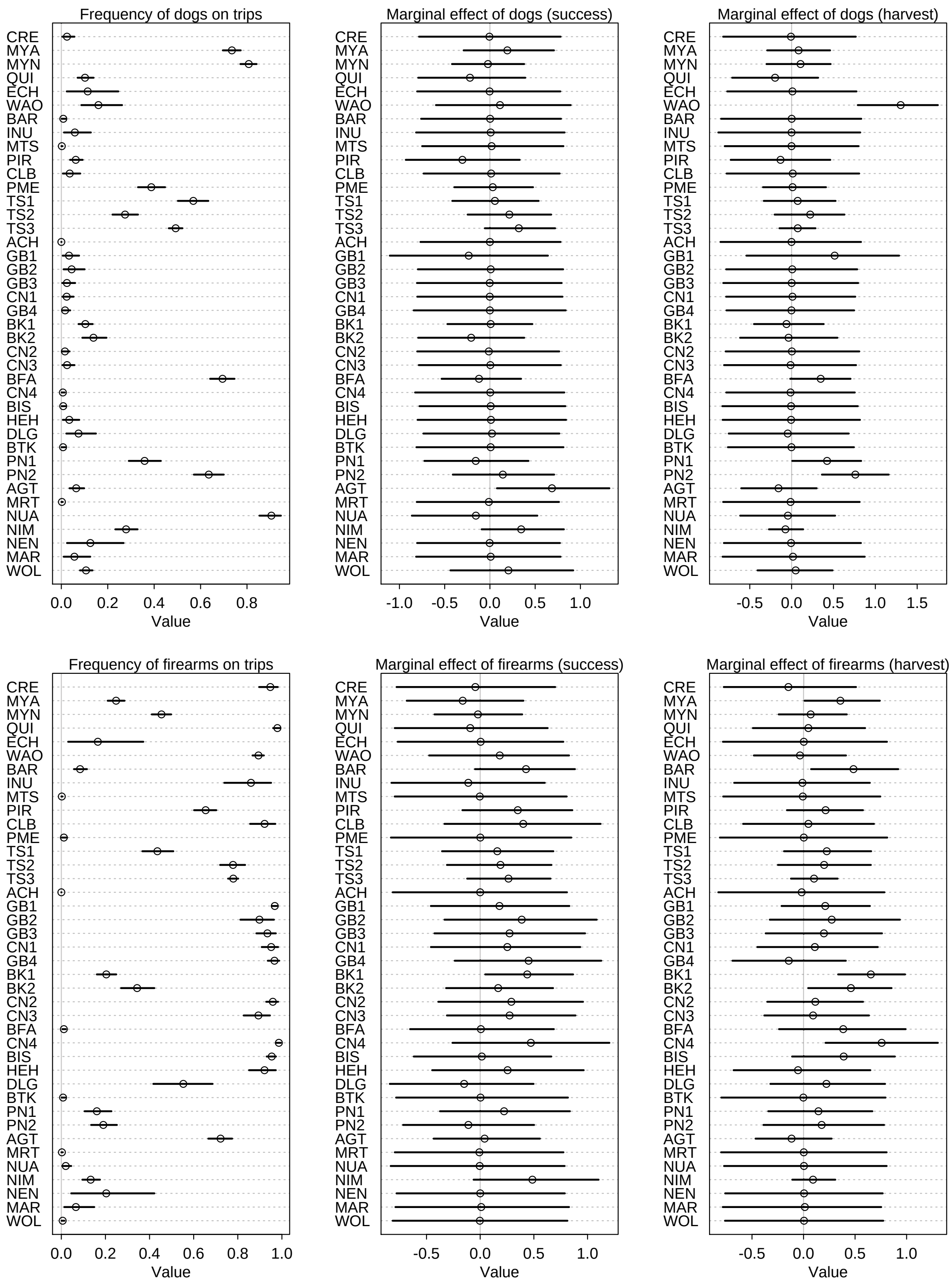

Figure S11. Marginal posterior distributions for dogs (top row) and firearms (bottom row). In the code, these parameters are named `dogs_mu`, `bdogs[1]`, `bdogs[2]`, `firearms_mu`, `bfirearms[1]`, and `bfirearms[2]`, respectively. Marginal distributions centered on zero with standard deviation 0.5 correspond to the prior. In those cases, the society contained no information to inform the parameter. Dogs are used at two sites, MTS and HEH, in which their use on trips was not documented. These missing data were averaged into the intercept and set to zero in this figure.

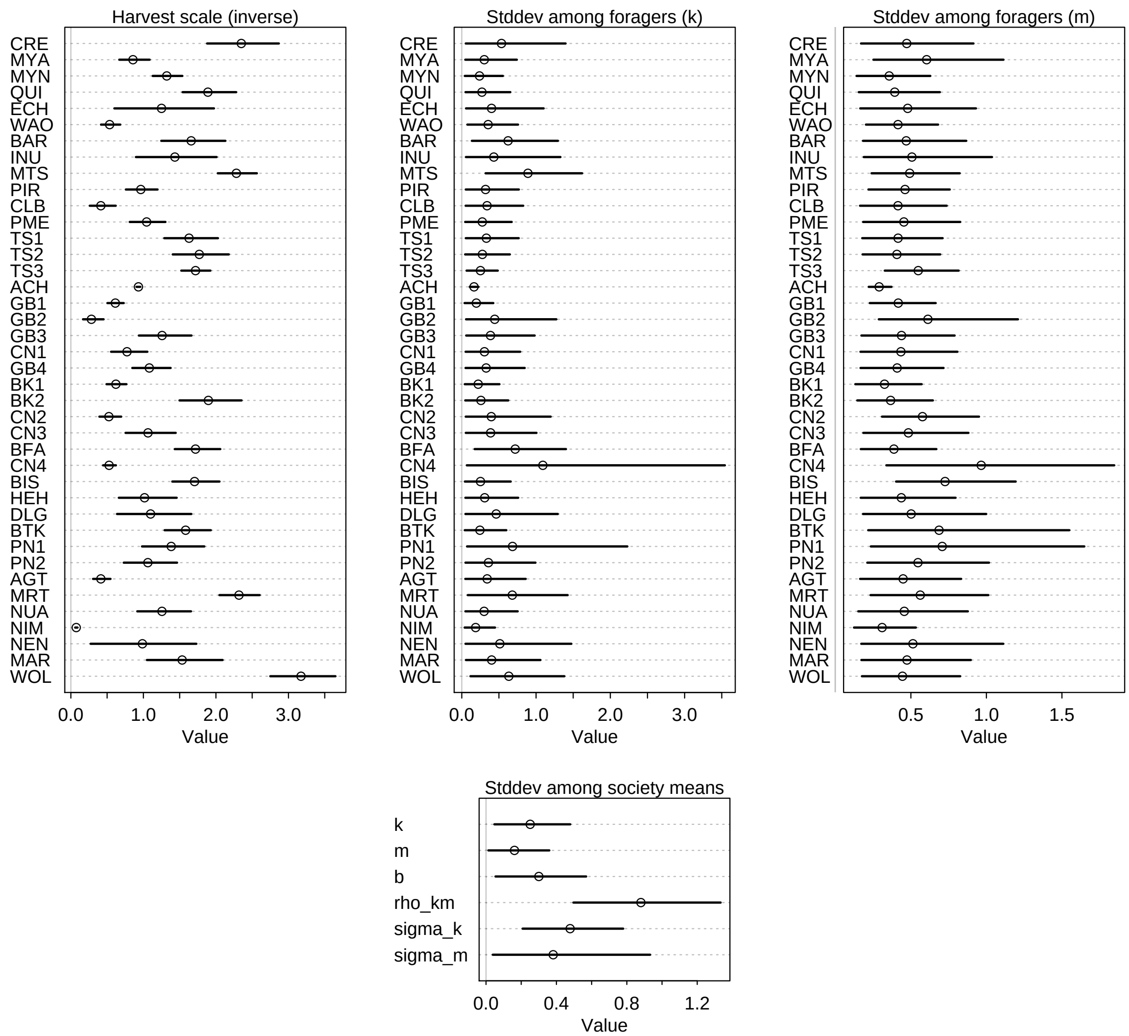

FIGURE S12. Marginal posterior distributions for dispersion and variance components. In the code, these parameters are named `hscale` (top-left), `sigmas_hunters[1]` (top-middle), `sigmas_hunters[2]` (top-right), and `sigma_societies` (bottom-middle). In the bottom-middle, k indicates the standard deviation among sites in mean skill growth, m the standard deviation among sites in mean skill decay, b the standard deviation in b across sites, `rho_km` the standard deviation (on the latent scale) of the correlations between k and m across sites, and then `sigma_k` and `sigma_m` are standard deviations across sites of standard deviations among foragers in each site.

Appendix A

Detailed model code. The full code for the model is available in the accompanying R package and scripts (https://osf.io/2kzb6/). In this section, we explain the model block of the code, focusing on how the marginalization over missing values is accomplished.

The first portion of the model block defines local variables, used in calculations, and priors. The only unusual code here is the Jacobian adjustment applied to `lifehistmeans[4]` and `lifehistmeans[5]`. This adjustment allows us to apply the prior on the natural, instead of logarithmic, scale.

```
model{
    // temp variables
    real k[N_hunters];
    real m[N_hunters];
    real b[N_societies];
    vector[N_trips] lm_f;
    vector[N_trips] lm_h;
    real p;
    real mu;
    matrix[2,2] Sigma;
    vector[N_trips] trip_duration_merge;

    // priors
    // society-level life history means --- centered on global means
    // equivalent to:
    //vs~multi_normal(lifehistmeans,quad_form_diag(Rho_societies,sigma_societies));
    //see transformation in transformed parameters block
    to_vector(zs) ~ normal(0,1);

    lifehistmeans[1:2] ~ normal( 1, prior_scale ); // log k,m
    lifehistmeans[3] ~ normal( 1, prior_scale ); // log b
    lifehistmeans[6] ~ normal( 0, prior_scale ); // shifted logit rho_km
    // do prior for stddev k,m between [4,5] as normal on transformed scale
    // this allows us to define same prior for sigma_societies[1:2]
    exp(lifehistmeans[4]) ~ normal( 0 , prior_scale );
    exp(lifehistmeans[5]) ~ normal( 0 , prior_scale );
    // need Jacobian adjustments for these priors
    // log|d/dy exp y| = log|exp y| = y
    // see also section 33.2 of Stan reference manual
    target += lifehistmeans[4];
    target += lifehistmeans[5];

    sigma_societies[1:3] ~ normal( 0 , prior_scale2 );
    sigma_societies[4] ~ normal( 0.5 , prior_scale2 );
    sigma_societies[5:6] ~ normal( 0 , prior_scale2 );

    dogs_mu ~ beta(2,10); // weighted to stop mode switching in site 8
    guns_mu ~ beta(2,4);
```

```
    ache_fix_rho ~ normal( 0, prior_scale );

    afbar ~ normal(0, prior_scale );
    ahbar ~ normal(0, prior_scale );
    sigma_af ~ normal(0, prior_scale );
    sigma_ah ~ normal(0, prior_scale );

    for ( s in 1:N_societies ) {
        af[1,s] ~ normal(afbar,sigma_af);
        ah[1,s] ~ normal(ahbar,sigma_ah);
        for ( i in 2:4 ) {
            af[i,s] ~ normal(0,prior_scale);
            ah[i,s] ~ normal(0,prior_scale);
        }
        sef[s] ~ normal(0,prior_scale);
        seh[s] ~ normal(0,prior_scale);
        for ( i in 1:2 ) {
            b_hours[i,s] ~ normal(0,prior_scale);
            b_dogs[i,s] ~ normal(0,prior_scale);
            b_firearms[i,s] ~ normal(0,prior_scale);
            se_dogs[i,s] ~ normal(0,prior_scale);
            se_firearms[i,s] ~ normal(0,prior_scale);

            b_xday[s,i] ~ normal(0,prior_scale);
        }
    }//s
    hscale ~ normal( 1 , prior_scale );

    // varying effects
    // foragers --- these are zero-centered
    // see translation to vh in transformed parameters block
    to_vector(zh) ~ normal(0,1);
```

The next chunk of code handles imputation of missing ages and trip durations. For each missing age, there is a corresponding standard error of the age. This comprises a Gaussian prior for the error of each missing age. Combined with the prior for each missing age, this provides a way to average over the uncertainty. For each missing trip duration, similarly a parameter is used. Then a vector that merges observed and missing values is generated. The prior formed from each site's (standardized) trip durations constrains the imputed values.

```
// age imputation
for ( i in 1:N_hunters ) {
    if ( age_impute_idx[i] > 0 ) {
        if ( age_impute_table[i,1]==1 )
            age_err[age_impute_idx[i]] ~
                    normal( 0 , age_impute_table[i,3] );
    }
```

```
}

// trip durations
for ( j in 1:N_societies ) trip_duration_mu[j] ~ normal(0,1);
trip_duration_sigma ~ exponential(1);
for ( i in 1:N_trips ) {
    if ( trip_hours[i]<0 ) {
        // missing
        trip_duration_merge[i] = trip_duration_imputed[hours_miss_idx[i]];
    } else {
        // observed
        trip_duration_merge[i] = log(trip_hours[i]);
    }
    // prior (when missing) or likelihood (when observed)
    trip_duration_merge[i] ~ normal( trip_duration_mu[trip_soc_id[i]] ,
                                     trip_duration_sigma[trip_soc_id[i]] );
}//i
```

The next short section computes hunter-specific and society-specific skill parameters. These are then reused in the likelihood calculations to follow.

```
// prep hunter effects so can re-use
for ( j in 1:N_hunters ) {
    k[j] = exp( lifehistmeans[1] + vs[forager_soc_id[j],1] + vh[j,1] );
    m[j] = exp( lifehistmeans[2] + vs[forager_soc_id[j],2] + vh[j,2] );
}
// prep b for each society, so only have to compute once
for ( s in 1:N_societies ) {
    b[s] = exp( lifehistmeans[3] + vs[s,3] ); // ensure positive with log link
}
```

The main loop of the model block comes next. This loop passes over trips, and then harvests within trips. The first chunk of code just prepares local variables. The `xdogsvec` and `xgunsvec` arrays exist to help us construct marginal log-probabilities when both dogs and firearms are unobserved (missing). The relevant code appears later down.

```
// likelihoods
lm_f = rep_vector(0,N_trips);
lm_h = lm_f;
// loop over trips and compute likelihoods
for ( i in 1:N_trips ) {
    real skillj;
    real sefx;
    real sehx;
    real ai;
    int hid;
    real avg_skill;
    vector[2] LLterms;
    vector[4] LL4terms;
    int xdogs;
```

```
    int xguns;
    int n_foragers_index;
    int coopidx;
    // prep binary tree for possible combinations of missing values
    int xdogsvec[4];
    int xgunsvec[4];
    xdogsvec[1] = 1;
    xdogsvec[2] = 1;
    xdogsvec[3] = 0;
    xdogsvec[4] = 0;
    xgunsvec[1] = 1;
    xgunsvec[2] = 0;
    xgunsvec[3] = 1;
    xgunsvec[4] = 0;
```

Next, when a trip has a pooled harvest, average skill for the entire group of hunters must be calculated. This is because we assume that production depends upon average skill in this case, where we cannot identify individual contributions. The `coopidx` variable tells us later which intercept parameter is needed, as the intercept in production differs depending upon pooled or individual harvests.

```
// compute avg skill (when needed)
avg_skill = 0;
if ( trip_pooled[i]==1 ) {
    // pooled harvest
    // compute average skill in foraging group
    for ( j in 1:n_foragers[i] ) {
        hid = forager_ids[i,j];
        if ( age_impute_idx[hid]==0 ) {
            // simple case, just fetch observed age
            ai = forager_age[i,j]; // from trip variables
        } else {
            // need some kind of imputation
            ai = forager_age[i,j] + age_err[age_impute_idx[hid]];
        }
        ai = ai/ref_age;
        skillj = exp(-m[hid]*ai)*pow(1-exp(-k[hid]*ai),b[trip_soc_id[i]]);
        avg_skill = avg_skill + skillj;
    }//j
    avg_skill = avg_skill/n_foragers[i] + 0.001;
    n_foragers_index = 1; // loop over just "one" forager
    coopidx = 3;
} else {
    // independent harvests
    n_foragers_index = n_foragers[i];
    coopidx = 2;
}
```

The big loop over individual foragers comes next. The loop begins by calculating individual forager skill, but only when harvest is not pooled. This code is structural the same as that used above to compute average skill, but it omits the averaging.

```
for ( j in 1:n_foragers_index ) {
    // if trip pooled, only one harvest (n_foragers_index==1)
    // otherwise loops over each harvest and predicts each

    if ( trip_pooled[i]==1 ) {
        skillj = avg_skill;
    } else {
        hid = forager_ids[i,j];
        if ( age_impute_idx[hid]==0 ) {
            // simple case, just fetch observed age
            ai = forager_age[i,j]; // from trip variables
        } else {
            // need some kind of imputation
            ai = forager_age[i,j] + age_err[age_impute_idx[hid]];
        }
        ai = ai/ref_age;
        skillj = exp(-m[hid]*ai)*pow(1-exp(-k[hid]*ai),b[trip_soc_id[i]]) + 0.001;
    }
```

Next we build "stem" expressions for each harvest log-probability. These stems contain all terms except those for dogs and firearms. Dogs and firearms must be added conditional on missingness. In that case, these stems are reused for each missingness state.

```
// failures production
lf_stem = exp( af[1,trip_soc_id[i]] +
                    af[coopidx,trip_soc_id[i]]*(n_foragers[i]-1) +
                    af[4,trip_soc_id[i]]*n_assistants[i,j] +
                    b_xday[trip_soc_id[i],1]*trip_xday[i]
                ) *
                exp(trip_duration_merge[i])^b_hours[1,trip_soc_id[i]];
// harvests production
lh_stem = exp( ah[1,trip_soc_id[i]] +
                    ah[coopidx,trip_soc_id[i]]*(n_foragers[i]-1) +
                    ah[4,trip_soc_id[i]]*n_assistants[i,j] +
                    b_xday[trip_soc_id[i],2]*trip_xday[i]
                ) *
                exp(trip_duration_merge[i])^b_hours[2,trip_soc_id[i]];
// failures skill elasticity
sef_stem = exp( sef[trip_soc_id[i]] );
// harvests skill elasticity
seh_stem = exp( seh[trip_soc_id[i]] );
```

Now we can do target updates. Different expressions need to be built, depending upon whether dogs, firearms, or both are missing. The simplest case is when both are observed. In this case, we just add the observed values to the stems, compute probability of failure, average harvest, and update. Note that -1 as the missingness indicator is chosen during data initialization. Note that the code here considers the probability of a zero harvest, instead of the probability of a non-zero harvest. This is equivalent to the analytical model definition given earlier, even though the expression looks different.

```
if ( n_dogs[i,j] != -1 && n_firearms[i,j] != -1 ) {
    // dogs and guns both observed
    // use obs values to update base rates of dogs and guns
    n_dogs[i,j] ~ bernoulli(dogs_mu[trip_soc_id[i]]);
    n_firearms[i,j] ~ bernoulli(guns_mu[trip_soc_id[i]]);
    // build production functions with observed values
    lm_f[i] = lf_stem * exp( b_dogs[1,trip_soc_id[i]]*n_dogs[i,j] +
                             b_firearms[1,trip_soc_id[i]]*n_firearms[i,j] );
    lm_h[i] = lh_stem * exp( b_dogs[2,trip_soc_id[i]]*n_dogs[i,j] +
                             b_firearms[2,trip_soc_id[i]]*n_firearms[i,j] );
    sefx = sef_stem * exp( se_dogs[1,trip_soc_id[i]]*n_dogs[i,j] +
                       se_firearms[1,trip_soc_id[i]]*n_firearms[i,j] );
    sehx = seh_stem * exp( se_dogs[2,trip_soc_id[i]]*n_dogs[i,j] +
                    se_firearms[2,trip_soc_id[i]]*n_firearms[i,j] );
    // compute failure probability and harvest mean
    p = 2*(1 - inv_logit( skillj^sefx * lm_f[i] ));
    mu = lm_h[i] * skillj^sehx;
    if ( trip_harvests[ i , j ]==0 )
        // failure
        1 ~ bernoulli(p);
    else {
        // observed harvest
        0 ~ bernoulli(p);
        trip_harvests[ i , j ] ~ gamma( mu/hscale[trip_soc_id[i]] ,
                                       1/hscale[trip_soc_id[i]] );
    }
}
```

The next two cases are when either dogs or firearms are missing. In these cases, we need to marginalize over missingness states. This generates two log-probability terms in a mixture.

```
// now dogs missing, guns observed
if ( n_dogs[i,j] == -1 && n_firearms[i,j] != -1 ) {
    n_firearms[i,j] ~ bernoulli(guns_mu[trip_soc_id[i]]);
    // average over missingness
    // LLterms holds terms to mix over
    // LLterms[1] is where dogs == 0
    // LLterms[2] is where dogs == 1
    for ( nterm in 1:2 ) {
        xdogs = nterm-1;
        lm_f[i] = lf_stem * exp( b_dogs[1,trip_soc_id[i]]*xdogs +
                                 b_firearms[1,trip_soc_id[i]]*n_firearms[i,j] );
        lm_h[i] = lh_stem * exp( b_dogs[2,trip_soc_id[i]]*xdogs +
                                 b_firearms[2,trip_soc_id[i]]*n_firearms[i,j] );
        sefx = sef_stem * exp( se_dogs[1,trip_soc_id[i]]*xdogs +
```

```
                        se_firearms[1,trip_soc_id[i]]*n_firearms[i,j] );
        sehx = seh_stem * exp( se_dogs[2,trip_soc_id[i]]*xdogs +
                        se_firearms[2,trip_soc_id[i]]*n_firearms[i,j] );
        p = 2*(1 - inv_logit( skillj^sefx * lm_f[i] ));
        mu = lm_h[i] * skillj^sehx;
        LLterms[nterm] = 0;
        if ( trip_harvests[i,j]==0 ) {
            LLterms[nterm] = LLterms[nterm] + log(p);
        } else {
            LLterms[nterm] = LLterms[nterm] + log1m(p);
            LLterms[nterm] = LLterms[nterm] +
                gamma_lpdf( trip_harvests[i,j] |
                    mu/hscale[trip_soc_id[i]] , 1/hscale[trip_soc_id[i]] );
        }
    }// nterm
    // do the mixture
    // Pr(dogs==1)*Pr(harvest|dogs==1) + Pr(dogs==0)Pr(harvest|dogs==0)
    // log_mix here is for numerical stability
    target += log_mix( dogs_mu[trip_soc_id[i]] , LLterms[2] , LLterms[1] );
}
// now dogs observed but firearms missing
if ( n_dogs[i,j] != -1 && n_firearms[i,j] == -1 ) {
    n_dogs[i,j] ~ bernoulli(dogs_mu[trip_soc_id[i]]);
    // average over missingness
    // similar to above, but LLterms now average over missing guns
    for ( nterm in 1:2 ) {
        xguns = nterm-1;
        lm_f[i] = lf_stem * exp( b_dogs[1,trip_soc_id[i]]*n_dogs[i,j] +
                              b_firearms[1,trip_soc_id[i]]*xguns );
        lm_h[i] = lh_stem * exp( b_dogs[2,trip_soc_id[i]]*n_dogs[i,j] +
                              b_firearms[2,trip_soc_id[i]]*xguns );
        sefx = sef_stem * exp( se_dogs[1,trip_soc_id[i]]*n_dogs[i,j] +
                        se_firearms[1,trip_soc_id[i]]*xguns );
        sehx = seh_stem * exp( se_dogs[2,trip_soc_id[i]]*n_dogs[i,j] +
                        se_firearms[2,trip_soc_id[i]]*xguns );
        p = 2*(1 - inv_logit( skillj^sefx * lm_f[i] ));
        mu = lm_h[i] * skillj^sehx;
        LLterms[nterm] = 0;
        if ( trip_harvests[i,j]==0 ) {
            LLterms[nterm] = LLterms[nterm] + log(p);
        } else {
            LLterms[nterm] = LLterms[nterm] + log1m(p);
            LLterms[nterm] = LLterms[nterm] +
                gamma_lpdf( trip_harvests[i,j] |
                    mu/hscale[trip_soc_id[i]] , 1/hscale[trip_soc_id[i]] );
        }
```

```
    }//nterm
    // do the mixture
    target += log_mix( guns_mu[trip_soc_id[i]] , LLterms[2] , LLterms[1] );
}
```

Finally, both dogs and firearms could be missing. In this case, we need a mixture over four possible states.

```
// finally, both dogs and guns missing
if ( n_dogs[i,j] == -1 && n_firearms[i,j] == -1 ) {
    // L4terms holds combinations of possible values of dogs and guns
    //     dogs guns {probability at site k}
    // [1] 1    1    dogs_mu[j] * guns_mu[k]
    // [2] 1    0    dogs_mu[j] * ( 1 - guns_mu[k] )
    // [3] 0    1    ( 1 - dogs_mu[j] ) * guns_mu[k]
    // [4] 0    0    ( 1 - dogs_mu[j] ) * ( 1 - guns_mu[k] )
    for ( nterm in 1:4 ) {
        xdogs = xdogsvec[nterm];
        xguns = xgunsvec[nterm];
        lm_f[i] = lf_stem * exp( b_dogs[1,trip_soc_id[i]]*xdogs +
                                 b_firearms[1,trip_soc_id[i]]*xguns );
        lm_h[i] = lh_stem * exp( b_dogs[2,trip_soc_id[i]]*xdogs +
                                 b_firearms[2,trip_soc_id[i]]*xguns );
        sefx = sef_stem * exp( se_dogs[1,trip_soc_id[i]]*xdogs +
                           se_firearms[1,trip_soc_id[i]]*xguns );
        sehx = seh_stem * exp( se_dogs[2,trip_soc_id[i]]*xdogs +
                           se_firearms[2,trip_soc_id[i]]*xguns );
        p = 2*(1 - inv_logit( skillj^sefx * lm_f[i] ));
        mu = lm_h[i] * skillj^sehx;
        LL4terms[nterm] = 0;
        if ( trip_harvests[i,j]==0 ) {
            LL4terms[nterm] = LL4terms[nterm] + log(p);
        } else {
            LL4terms[nterm] = LL4terms[nterm] + log1m(p);
            LL4terms[nterm] = LL4terms[nterm] +
                 gamma_lpdf( trip_harvests[i,j] |
                     mu/hscale[trip_soc_id[i]] ,
                     1/hscale[trip_soc_id[i]] );
        }
        // add leading factor for probability of combination of missingness
        if ( xdogs==1 )
            LL4terms[nterm] = LL4terms[nterm] + log(dogs_mu[trip_soc_id[i]]);
        else
            LL4terms[nterm] = LL4terms[nterm] + log1m(dogs_mu[trip_soc_id[i]]);
        if ( xguns==1 )
            LL4terms[nterm] = LL4terms[nterm] + log(guns_mu[trip_soc_id[i]]);
        else
            LL4terms[nterm] = LL4terms[nterm] + log1m(guns_mu[trip_soc_id[i]]);
```

```
    }//nterm
    // do the mixture
    target += log_sum_exp( LL4terms );
}
```

In the end, the model block just loops over foragers j and trips i until all trips have been processed.

```
        } //j over foragers
    } //i over trips
} //model
```